

## BBCode

### Kurzbeschreibung

Ersetzung von BBCode-Tags und Smilies in Texten. Eine Übersicht über die verwendbaren BBCodes und Smilies befindet sich in der Datei `infos/bbcode.php`.

### Einbindung

```
require_once(PATH_TO_ROOT."common/classes/class.bbcode.inc.php");
```

### Instanziierung

Mit Konstruktor `BBCode([bool $htmlEntities[, bool $images[, bool $smilies[, bool $code]])`.

`$htmlEntities` Im Text werden Sonderzeichen nicht mit [htmlentities\(\)](#) maskiert, wenn false übergeben wird (z.B. wenn der Text schon maskiert ist). Standard ist true.

`$images` Wird hier false übergeben, werden `[img]`-Tags zur Darstellung von Bildern nicht umgewandelt. Standard ist true.

`$smilies` Bei der Übergabe von false werden Smilies nicht durch Icons ersetzt. Standard ist true.

`$code` Gibt an, ob BBCodes überhaupt ersetzt werden sollen (schließt den `[img]`-Tag ein). Standard ist true.

### string parse(string \$text)

Parst einen Text, ersetzt Codes und Smilies gemäß den im Konstruktor gemachten Einstellungen und liefert HTML-Code zurück.

### string stripTags(string \$text)

Entfernt BBCode-Tags (`[img]`-Tags samt URL) aus dem übergebenen Text. Hier ist ebenfalls der Aufruf über ein Objekt nötig. Ein statischer Aufruf funktioniert nicht.

### string getTagOverviewLink([string \$title])

Gibt den HTML-Code für einen Link zurück, der auf die Übersichts-Datei `infos/bbcode.php` verweist. Der Titel des Links ist standardmäßig „Formatierungsmöglichkeiten“.

Hier ist auch ein statischer Aufruf möglich.

## Data

### Kurzbeschreibung

Funktionen zur sicheren Ausgabe von Daten an den Browser bzw. in SQL-Strings.

Hier werden nur statische Methoden erwähnt, für die erweiterten Funktionen der Klasse siehe Datei `common/classes/class.data.inc.php`.

### Einbindung

Nicht nötig, wird durch `common/init.inc.php` automatisch überall eingebunden.

### Instanziierung

Nicht nötig, Zugriff auf die statischen Methoden erfolgt mit `Data::<Methode>`.

### **string toHTML(string \$data[, bool \$isGPCData])**

---

Maskiert Sonderzeichen im übergebenen String mittels [htmlentities\(\)](#) und der korrekten Zeichenkodierung zur sicheren Ausgabe im Browser.

\$isGPCData gibt an, ob die Daten aus \$\_GET, \$\_POST oder \$\_COOKIE stammen. Ist das der Fall, werden vorher Backslashes zum Escapen von Zeichen wie ', " oder \ entfernt. Standard ist hier true, muss aber in der Regel auf false gesetzt werden, wenn die Daten zur Ausgabe z.B. aus der Datenbank kommen.

### **string toMysql(string \$data[, bool \$isGPCData])**

---

Maskiert Sonderzeichen im übergebenen String mittels [mysql\\_real\\_escape\\_string\(\)](#) zur sicheren Verwendung in einem SQL-String.

\$isGPCData gibt an, ob die Daten aus \$\_GET, \$\_POST oder \$\_COOKIE stammen. Ist das der Fall, werden vorher Backslashes zum Escapen von Zeichen wie ', " oder \ entfernt. Standard ist hier true.

### **string gpcUnescape(string \$data)**

---

Entfernt Backslashes mittels [stripslashes\(\)](#) zum Escapen von Zeichen wie ', " oder \ in Strings, die aus \$\_GET, \$\_POST oder \$\_COOKIE stammen, wenn der Webserver gemäß seinen Einstellungen dieses Escape-Zeichen eingefügt hat.

Die Funktion wird auch von den beiden oben erwähnten Funktionen verwendet.

## **DateSelection**

### **Kurzbeschreibung**

Datumsauswahl mit Dropdown-Menüs.

### **Einbindung**

```
require_once(PATH_TO_ROOT."common/classes/class.dateselection.inc.php");
```

### **Instanziierung**

Mit Konstruktor `DateSelection(int $tag, int $monat, int $jahr[, int $stunde[, int $minute]])`.

Anzugeben sind die voreingestellten Werte. Standard für Stunde und Minute ist 0, falls nicht angegeben.

Für eine Beschreibung der Klassenmethoden siehe Datei `common/classes/class.dateselection.inc.php` oder die phpDoc-Dokumentation im CVS.

## **Email**

### **Kurzbeschreibung**

Versenden von Emails aus dem Portal.

### **Einbindung**

```
require_once(PATH_TO_ROOT."common/classes/class.email.inc.php");
```

### **Instanziierung**

Mit Konstruktor `Email(int $userID)`.

Anzugeben ist die Benutzer-ID des Absenders.

**bool sendMessageToUser(string \$subject, mixed \$to, string \$message[, bool \$single])**

---

Email an einen oder mehrere User des Portals senden. Gibt true bei Erfolg zurück, sonst false.

\$subject    Betreff

\$to            ID des Empfängers (Integer) oder Array von Integeren für mehrere Empfänger

\$message    Nachricht

\$single      Einzelversand an mehrere Empfänger (false, Standard) oder nur eine Email für alle Empfänger (true)

**bool sendMessageToAddress(string \$subject, mixed \$to, string \$message[, bool \$single])**

---

Email an einen oder mehrere Email-Adressen senden. Gibt true bei Erfolg zurück, sonst false.

\$subject    Betreff

\$to            Adresse des Empfängers (String) oder Array von Strings für mehrere Empfänger

\$message    Nachricht

\$single      Einzelversand an mehrere Empfänger (false, Standard) oder nur eine Email für alle Empfänger (true)

**bool sendMessageToAddressFromNotUser(string \$subject, string \$from, mixed \$to, string \$message[, bool \$single])**

---

Email an einen oder mehrere Email-Adressen von einem nicht am Portal registrierten Benutzer senden. Gibt true bei Erfolg zurück, sonst false.

\$subject    Betreff

\$from        Absenderadresse

\$to            Adresse des Empfängers (String) oder Array von Strings für mehrere Empfänger

\$message    Nachricht

\$single      Einzelversand an mehrere Empfänger (false, Standard) oder nur eine Email für alle Empfänger (true)

**bool validate(string \$address)**

---

Überprüft die angegebene Email-Adresse auf syntaktische Richtigkeit. Gibt true zurück, falls die Email-Adresse korrekt ist, sonst false.

**void sendSignature(bool \$how)**

---

Standardmäßig wird an alle Emails die eStudy-Signatur angehängt. Wird diese Funktion vorher mit dem Parameter false aufgerufen, wird die Signatur abgeschaltet.

**void setContentype(string \$show)**

---

Emails werden standardmäßig im Plaintext-Format versandt. Mit dieser Funktion kann das Format eingestellt werden:

„txt“ für Plaintext-Format  
„html“ für HTML-Format

<b>Output</b>
---------------

**Kurzbeschreibung**

Diverse Ausgabefunktionen.

**Einbindung**

Nicht nötig, wird durch common/init.inc.php automatisch überall eingebunden.

**Instanziierung**

Nicht nötig, Zugriff auf die statischen Methoden erfolgt mit `Output::<Methode>`.

---

**string echoDate([string \$picture[, mixed \$timestamp]])**

---

Gibt ein formatiertes Datum unter Berücksichtigung der Zeitzoneneinstellungen und Sommer-/ Winterzeit zurück.

`$picture`      Formatstring wie bei PHP [date\(\)](#)-Funktion. Standard ist „d.m.Y H:i“.

`$timestamp`   Entweder Unix-Timestamp (Integer, eventuell Typecast verwenden!) oder  
Datumstring. Wird der Parameter nicht angegeben, wird die aktuelle Zeit  
zurückgegeben.

---

**void echoTableHead(string \$title[, int \$colspan])**

---

Ausgabe der Kopfzeile einer Tabelle.

Etwa: `<tr><td class='tableHead' colspan='$colspan'>$title</td></tr>`  
`$colspan` ist standardmäßig 1.

---

**void echoMessage(string \$message[, int \$colspan])**

---

Gibt eine Tabellenzeile mit einer Nachricht (z.B. Erfolgsmeldung) aus.

Etwa: `<tr><td class='message' colspan='$colspan'>$message</td></tr>`  
`$colspan` ist standardmäßig 1.

---

**void errorMessage(string \$message[, int \$colspan[, bool \$fatal[, bool \$table]]])**

---

Gibt eine Tabellenzeile mit einer Fehlermeldung aus.

Etwa: `<tr><td class='error' colspan='$colspan'>$message</td></tr>`  
`$colspan` ist standardmäßig 1.

`$fatal`      Wird hier true übergeben, wird zusätzlich zu `$message` noch folgender Text ausgegeben:  
„Bitte wenden Sie sich an den Webmaster, falls der Fehler noch einmal auftritt: <Email>“. Standardmäßig false.

`$table`      Bei true wird die Zeile mit `<table>`-Tags eingefasst. Standardmäßig false.

---

**bool confirmQuestion(string \$callingscript, string \$text[, int \$width])**

---

Gibt eine Bestätigungsfrage aus (mit den Buttons 'Ja' und 'Nein'), ruft das Script wieder auf und setzt es an der selben Stelle fort.

Beispiel:

```
$confirmed = Output::confirmQuestion(PATH_TO_ROOT.SCRIPT_NAME,
"Wollen Sie das wirklich?");
if ($confirmed === true) {
    ...
} elseif ($confirmed === false) {
    ...
}
```

Mit dem Parameter \$width kann die Breite der Tabelle in Prozent angegeben werden.

---

### **void mailTo(int \$userID[, string \$linklabel])**

---

Gibt einen Link auf das Mailformular aus. Der voreingestellte Empfänger ist der durch \$userID bestimmte Benutzer. Wird \$linklabel nicht angegeben, werden Vor- und Nachname des Benutzers als Linktext verwendet.

---

### **void linkToProfile(int \$userID[, string \$linklabel])**

---

Gibt einen Link zum Profil des durch \$userID bestimmten Benutzers aus. Wird \$linklabel nicht angegeben, werden Vor- und Nachname des Benutzers als Linktext verwendet.

---

### **void echoDialogBottom([array \$labels[, int \$colspan[, mixed \$resettype[, string \$additional]]]])**

---

Gibt die Fußzeile eines Formulars aus.

\$labels      Assoziatives Array mit den Beschriftungen der drei Standardbuttons. Standardmäßig:  
              okButton => „OK“  
              resetButton => „Zurücksetzen“  
              cancelButton => „Abbrechen“

\$colspan      colspan der Tabellenzeile, standardmäßig 1.

\$resettype    „Zurücksetzen“-Button erzeugen als:  
              true: reset (Standard)  
              false: submit  
              „none“: nicht anzeigen

\$additional    Zwischen OK- und Reset-Button auszugebender Text, z.B. zusätzliche Buttons.

---

### **string getUsergroupPicture(int \$userID, int \$usergroup[, int \$courseID])**

---

Gibt den HTML-Code zur Anzeige des Icons zur Benutzergruppe eines Benutzers zurück.

\$userID      Benutzer-ID

\$usergroup    Benutzergruppe

\$courseID    Kurs-ID. Benutzer können in Kursen u.U. eine andere Benutzergruppe haben als im Foyer.

---

### **string getIcon(string \$classname, string \$title)**

---

Gibt den HTML-Code zur Anzeige eines Icons zurück. \$classname bestimmt den Namen der CSS-Klasse, die auf das Icon angewendet wird. Die Klasse muss vorher in der CSS-Datei definiert worden sein. Der URL des Icons wird ebenfalls per CSS festgelegt. \$title gibt das alt- und title-Attribut des Icons an.

<b>Utilities</b>
------------------

#### **Kurzbeschreibung**

Diverse nützliche Funktionen.

#### **Einbindung**

Nicht nötig, wird durch common/init.inc.php automatisch überall eingebunden.

#### **Instanziierung**

Nicht nötig, Zugriff auf die statischen Methoden erfolgt mit `Utilities::<Methode>`.

---

### **string concatGetVars()**

---

Fügt die \$\_GET-Variablen zu einem String zusammen, der an eine URL angehängt werden kann. Rückgabe: String in der Form `?Var1=Value1&Var2=Value2&...`

---

### **bool mkdir\_r(string \$path[, int \$rights])**

---

Rekursive Version von [mkdir\(\)](#) zum Erstellen von mehreren Verzeichnisebenen mit einem Aufruf. Gibt true bei Erfolg zurück, sonst false.

\$path    Pfad, der erstellt werden soll.

\$rights   Zugriffsrechte, mit denen die Verzeichnisse erstellt werden. Für die gewohnte oktale Schreibweise muss die erste Ziffer eine 0 sein. Standard ist 0777. Siehe auch [chmod\(\)](#).

---

### **mixed unserializeSession(mixed \$id)**

---

Gibt die Daten einer Session in einem identisch aufgebauten assoziativen Array zurück.

Übergeben werden kann entweder die Session-ID (String) oder die ID des Benutzers, zu dem die Session gehört (Integer, eventuell Typecast verwenden!).

Im Fehlerfall (z.B. Session nicht vorhanden) wird false zurückgegeben.

---

### **bool userOnline(mixed \$id)**

---

Gibt true zurück, wenn der durch die Session-ID (String) oder User-ID (Integer, eventuell Typecast verwenden!) bestimmte Benutzer noch online ist, sonst false.

Ein Benutzer gilt als online, wenn er sich noch nicht ausgeloggt hat und seit seinem letzten Zugriff weniger als 15 Minuten vergangen sind.

---

### **bool validateURL(string \$url)**

---

Gibt true zurück, wenn der übergebene URL syntaktisch korrekt ist, sonst false.

**array setArrayIndexes(array \$array, string \$index)**

---

Erstellt aus einem zweidimensionalen Array ein neues Array, in dem die Werte des angegebenen Indexes der assoziativen zweiten Dimension des alten Arrays die Indexe der ersten Dimension des neuen Arrays sind.

Beispiel:

\$array\_a:

```
Array
(
    [0] => Array
        (
            [ID] => 8
            [Name] => Max
        )
    [1] => Array
        (
            [ID] => 34
            [Name] => Tim
        )
)
```

setArrayIndexes(\$array\_a, "ID") gibt folgendes Array zurück:

```
Array
(
    [8] => Array
        (
            [ID] => 8
            [Name] => Max
        )
    [34] => Array
        (
            [ID] => 34
            [Name] => Tim
        )
)
```

Die Reihenfolge der Array-Elemente wird dabei nicht verändert, ein Durchlauf mit foreach ist also weiterhin möglich.

Im Fehlerfall wird das unveränderte Ausgangsarray zurückgegeben.